MIE1619: Constraint Programming and Hybrid Optimization

Instructor: J. Christopher Beck
Office: BA8126 (meetings by appointment)
Email: jcb@mie.utoronto.ca Website: q.utoronto.ca

December 11, 2018

Lectures

Wednesday: 9:10 – 12:00, UC255, starting January 9.

Overview

The topic of MIE1619 is the "non-traditional" optimization technique Constraint Programming (CP) and hybrids of CP with approaches in OR. Heavy emphasis will be placed on similarities and differences between CP and mathematical programming including the unified framework of search, relaxation, and inference. The primary hybrid approaches will be based on constraint generation approaches including Logic-based Benders Decomposition and SAT Modulo Theory.

This is an advanced graduate level course intended for research-stream students. *MEng students are not admitted without special permission from the instructor.* The course will be challenging. Students are expected to read material in preparation for each lecture and, in a few cases, view online lectures.

An objective of this course is to impart skills necessary for an academic career such as paper writing, presentation skills, and writing peer reviews. The main evaluation will be a project where the student is expected to apply techniques discussed in the course to their own research interests: you should do something you weren't already planning to do as part of your research. A goal of this course is that these projects will be publishable in a peer-reviewed forum.

Evaluation

Component		**Value**	**Notes**
Modeling Assignments (3)		15	5 marks each
Peer Reviews (2)		20	10 marks each
Project		65	
Proposal	0		
Checkpoint	5		
Initial Submission	15		
Presentation	15		
Final Submission	30		
Total		100	

Modeling Assignments: Value 15%

One of the goals of this course is to be able to model combinatorial optimization problems in CP. To that end, there are three modeling assignments that are done in Weeks 2, 3, and 4 as follows:

1. **Due Date: Jan 15, midnight via Quercus. Value 5%** You are to watch Week 1 of the MOOC "Basic Modeling for Discrete Optimization" (Stuckey and Lee, 2017) and complete the Week 1 assignment for the MOOC. See Quercus site for more details of what you are to hand in.

2. **Due Date: Jan 29, midnight via Quercus. Value 5%.** Watch Week 2 of the same MOOC (Stuckey and Lee, 2017) and complete the Week 2 assignment. See Quercus site for more details of what you are to hand in.

3. **Due Date: Jan 30, in class. Value 5%.** At 9:10 AM, pairs of students will be given a problem to model and solve in the following two hours. In the last hour of lecture, each team will present their problem and solution.

Peer Reviews: Value 20%

Each student will complete two peer reviews as follows:

1. **Due Date: Mar 5, midnight via Quercus. Value 10%.** A conference submission will be provided to all students and they will write a peer review.

2. **Due Date: Mar 26, midnight via Quercus. Value 10%.** The initial submission of each project will be peer reviewed by a least two people: the instructor and a student. The identity of the reviewers is to be kept confidential but the identity of the author is not (i.e., we are following a single-blind reviewing process not a double-blind process).

Project: Value 65%

The project is a full research paper. It includes a literature survey and a novel contribution in constraint programming or hybrid techniques. Such a contribution will probably include empirical work (i.e., computational experiments) but could also be a mathematical or theoretical contribution. An ideal situation would be for the student to apply the course material to his/her own research. Failing this, consult with the instructor for project ideas. All topics must be approved by the instructor.

The project is to be done independently. In special cases, permission may be given to do the project in a pair. However, a pair project must have a specific design: the same problem will be attacked by at least two different approaches. The students in the pair must work independently on the approaches. The project has a number of components:

1. **Proposal. Due date: Feb 5, midnight via Quercus. Value: 0%.** One-page description of the goal of the project and the approach or approaches considered. The proposal must be approved by the instructor.

2. **Checkpoint. Due date: Feb 26, midnight via Quercus. Value: 5%.** The checkpoint contains a bibliography and preliminary literature review that summarizes the relevant existing work. In addition, the checkpoint should report the state of the project, any preliminary experiments that have been run, hypotheses that have been formed, and the plan for the rest of the project.

3. **Initial Submission. Due date: Mar 19, midnight via Quercus. Value: 15%.** The initial submission is a complete draft of the paper, equivalent to what you would submit for peer review. Students are encouraged to continue working on the project after the submission and are expected to make changes to the paper (possibly including further experimentation, etc.) in response to the peer reviews.

4. **Presentation: Due date: Apr 3 & 10, in class. Value: 15%.** The projects will be presented in a standard conference-style presentation. You will have a 25-minute time slot of which 20 minutes will be for the presentation and 5 minutes for questions. You need to present your work so that the audience understands the contributions you have made.

5. **Final Submission. Due date: Apr 12, midnight via Quercus. Value: 30%.** This is the final version of the paper. Evaluation will be based on the final submission as well as the changes that have been made from the initial submission (i.e., if you ignore reasonable comments in the reviews, you will lose marks). Marks will be based on content and writing.

Class Participation

There is no explicit mark for class participation. However, marks for any of the above may be adjusted to account for lack of participation. Students are expected to actively participate in all lectures by asking and answering questions, making suggestions, asking for and providing clarification of important concepts. Students are expected to have read the required readings for each lecture and to have made use of the additional material in areas where they are confused. The level of participation in the class will be used as a guide to the extent that the students have completed and understood the readings.

Lateness

All deadlines are strict. Submissions after the deadline will lose 25% per 24-hour period.

Software

There are a variety of free-for-academic-use software for Constraint Programming. You are free to choose and use whatever tools you want with the following considerations.

1. It must be legal for you to use the software.
2. You must use a CP solver. For your project you may want to *also* use a MIP solver but the requirements for the course are that you write CP models and solve them with a CP solver.
3. For the Modeling Bootcamp (see Week 4), you will be paired-up with another student. It is recommended that you both use the same platform.
4. The MOOC that you will be watching for Week 2 and 4 uses the Minizinc constraint language and IDE. You may want to take this into consideration in choosing a platform.

Possible choices (not an exhaustive list):

- Minizinc (www.minizinc.org)
- IBM ILOG OPL (on the ECFPC system) and available for download (see IBM Academic Initiative website)
- CPOptimizer (see IBM Academic Initiative website): C++
- Google OR Tools (code.google.com/p/or-tools/): C++, Python (?)

Primary Texts

We will be making substantial use of the following books (available at no cost through the library or online) which will be supplemented with recent papers as noted below.

- Hooker (2014): J.N. Hooker, *Integrated Methods for Optimization*.
  See http://lib.myilibrary.com.myaccess.library.utoronto.ca/Open.aspx?id=133666
- Dechter (2003): R. Dechter, *Constraint Processing*.
  See http://www.sciencedirect.com.myaccess.library.utoronto.ca/science/book/9781558608900.
- Rossi et al. (2006): F. Rossi, P. van Beek, T. Walsh, *The Handbook of Constraint Programming*.
  See http://www.sciencedirect.com.myaccess.library.utoronto.ca/science/bookseries/15746526/2
- Stuckey and Lee (2017): P. Stuckey, J. Lee, *Basic Modeling for Discrete Optimization*, Coursera MOOC.
  See https://www.coursera.org/learn/basic-modeling/home/welcome.
- Bergman et al. (2016): Bergman, D., A. A. Cire, W. van Hoeve, J. Hooker. 2016. *Decision Diagrams for Optimization*.
  See https://link-springer-com.myaccess.library.utoronto.ca/book/10.1007%2F978-3-319-42849-9.

Weekly Lectures

Note that this is subject to change. I do not anticipate any major changes (perhaps some changes in the readings), however for the up-to-date details, you should consult the weekly details document on Quercus.

Week	Topics	Tasks & Readings
1: Jan 9	Introduction	Dechter (2003), Ch 1
2: Jan 16	Search, Inference, and Relaxation Consistency in CP	Stuckey and Lee (2017) Week 1; Hooker (2014), Ch 1, 2 Hooker (2014) Ch 6.1, Dechter (2003) Ch 3
3: Jan 23	Global Constraints	Hooker (2014) Ch 6, Sect 7-10; Rossi et al. (2006) Ch 6 & 22
4: Jan 30	Modeling Bootcamp	Smith (2006); Stuckey and Lee (2017) Week 2
5: Feb 6	Dynamic Programming, Decision Diagrams, and Global Constraints	Trick (2003) or van Hoeve et al. (2006); Bergman et al. (2016) Ch 1 & 10
6: Feb 13	Search	Dechter (2003) Ch 5 Sect 1-3; Hooker (2014) Ch 5 Sect 1.0-1.5
7: Feb 27	OR/CP Hybrids & Peer Reviews	Laborie and Rogerie (2016), Achterberg et al. (2008), Benos et al. (2003)
8: Mar 6	Constraint Generation: Benders Decomposition	Hooker (2014) Ch 2.8, 5.2-5.2.2, Tran et al. (2016)
9: Mar 13	Satisfiability and MaxSAT	Hooker (2014) Ch 5.2.4-5, Gomes et al. (2008) up to end of Ch 2.2.5
10: Mar 20	Constraint Generation: SAT Modulo Theory and Lazy Constraint Generation	De Moura and Bjørner (2011), Schutt et al. (2013)
11: Mar 27	Large Neighborhood Search	Pisinger and Ropke (2010)
12: Apr 3	Project Presentations	
13: Apr 10	Project Presentations	

References

Achterberg, T., T. Berthold, T. Koch, K. Wolter. 2008. Constraint integer programming: A new approach to integrate CP and MIP. *Proceedings of the Fifth International Conference on the Integration of AI and OR Techniques in Constraint Programming (CPAIOR08)*. 6–20.

Benos, D.J., K.L. Kirk, J.E. Hall. 2003. How to review a paper. *Advances in Physiology Education* **27**(2) 47–52.

Bergman, D., A. A. Cire, W. van Hoeve, J. Hooker. 2016. *Decision Diagrams for Optimization*. Springer.

De Moura, L., N. Bjørner. 2011. Satisfiability modulo theories: Introduction and applications. *Communications of the ACM* **54**(9) 69–77.

Dechter, R. 2003. *Constraint Processing*. Morgan Kaufmann.

Gomes, C. P., H. Kautz, A. Sabharwal, B. Selman. 2008. Satisfiability solvers. F. van Harmelen, V. Lifschitz, B. Porter, eds., *Handbook of Knowledge Representation*. Elsevier B.V., 89–134.

Hooker, J. N. 2014. *Integerated Methods for Optimization*. 2nd ed. Springer.

Laborie, P., J. Rogerie. 2016. Temporal linear relaxation in IBM ILOG CP Optimizer. *J0urnal of Scheduling* **19** 391—-400.

Pisinger, D., S. Ropke. 2010. Large neighborhood search. M. Gendreau, J.-Y. Potvin, eds., *Handbook of Metaheuristics*. Springer Science+Business Media, LLC, 399–419.

Rossi, F., P. van Beek, T. Walsh, eds. 2006. *Handbook of Constraint Programming*. Elsevier.

Schutt, A., T. Feydy, P. J. Stuckey, M. Wallace. 2013. Solving RCPSP/max by lazy clause generation. *Journal of Scheduling* **16**(3) 273–289.

Smith, B. 2006. Modelling. F. Rossi, P. van Beek, T. Walsh, eds., *Handbook of Constraint Programming*, chap. 11. Elsevier, 377–406.

Stuckey, P. J., J. H. M. Lee. 2017. Basic modeling for discrete optimization. Coursera MOOC. Https://www.coursera.org/learn/basic-modeling/home/welcome.

Tran, T. T., A. Araujo, J. C. Beck. 2016. Decomposition methods for the parallel machine scheduling problem with setups. *INFORMS Journal on Computing* **28**(1) 83–95.

Trick, M. A. 2003. A dynamic programming approach for consistency and propagation for knapsack constraints. *Annals of Operations Research* **118**(73–84).

van Hoeve, W.-J., G. Pesant, L.-M. Rousseau, A. Sabharwal. 2006. Revisiting the sequence constraint. F. Benhamou, ed., *Proceedings of the Twelfth International Conference on the Principles and Practice of Constraint Programming (CP06)*. 620–634.